eRoute: Mobility-Driven Integration of Heterogeneous Urban Cyber-Physical Systems Under Disruptive Events

Yukun Yuan, *Student Member, IEEE*, Desheng Zhang, *Member, IEEE*, Fei Miao, *Member, IEEE*, John A. Stankovic, *Life Fellow, IEEE*, Tian He, *Fellow, IEEE*, George J. Pappas, *Fellow, IEEE*, and Shan Lin, *Senior Member, IEEE*

Abstract—With the rapid development of cities, heterogeneous urban cyber-physical systems are designed to improve citizens' experience, e.g., navigation and delivery service. However, the integration of services is not designed for disruptive events, an oversight that has rippling effects on service quality. For example, urban transportation systems consist of multiple transport modes that have complementary characteristics of capacities, speeds, and costs, facilitating smooth passenger transfers by planned schedules. Such integration may experience significantly increased delays during disruptions. Current solutions rely on a substitute service to transport passengers from and to affected areas using ad-hoc schedules and static routes, which are inefficient and do not utilize mobility patterns of mobile systems, e.g., dynamic passenger demand. To coordinate heterogeneous transportation systems under disruptions, we design a service to automatically select and integrate part of three systems (subway, bus, and taxi) using systems' mobility patterns, e.g., predicted supply and demand. The service is presented in a normal version, eRoute, considering both subway and bus, and in a version taking taxis into account, called enhanced eRoute. We implement and evaluate eRoute with datasets including subway, bus and taxi, and a fare collection system. The data-driven evaluation results show that eRoute improves the ratio of served passengers per time interval by up to 11.5 times and reduces the average traveling time by up to 82.1 percent compared with existing solutions.

Index Terms—Vehicles mobility management, schedules, passenger mobility, integrated networks, disruptions

1 Introduction

With rapid urbanization, a large number of urban mobile systems, e.g., smart card networks, ride-sharing systems, and safety surveillance systems, have been developed and incorporated into modern cities to improve the efficiency of urban service operations and citizen experiences. However, various disruptive events can interrupt the normal operation of the integrated urban systems and introduce unexpected service quality degradation. In this work, we take urban transportation systems as an example to investigate how to manage mobility across heterogeneous urban cyber-physical systems under disruptive events.

An urban transportation system typically consists of multiple systems, e.g., subway, bus, and taxi. These systems have complementary characteristics to meet the different needs of passengers. Specifically, the subway serves as the backbone of the urban transportation network, providing high-speed, high-capacity transport services across major areas; the bus is slower and cheaper, but it spreads over the entire city with a large number of different lines and stations; the taxi provides flexible pick-up and drop-off locations, also allowing flexible routes to adapt to traffic conditions.

In an integrated transportation system, heterogeneous transport modes are designed to connect by closely-located stations and synchronized schedules to facilitate passenger transfer activities under normal operations. However, under various disruptive events, that can cause stations or vehicles to shut down for an unpredictable period, e.g., a power failure or a signal error, the integrated heterogeneous transportation systems become disconnected and inefficient, resulting in a surge of stranded passengers and cascading delays in affected areas. Therefore, it is a very challenging problem to deal with such disruptive events in heterogeneous transportation systems.

There are very limited solutions to serve stranded passengers in the current transportation systems. Existing practices typically provide substitute services using backup vehicles, e.g., dispatching empty shuttles to the closed

- *Yukun Yuan and Shan Lin are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11790 USA. E-mail: {yukun.yuan, shan.x.lin}@stonybrook.edu.*
- *Desheng Zhang is with the Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA. E-mail: dz220@cs.rutgers.edu.*
- *Fei Miao is with the Department of Computer Science and Engineering, University of Connecticut, Storrs Mansfield, CT 06268 USA. E-mail: fei.miao@uconn.edu.*
- *John A. Stankovic is with the Department of Computer Science, University of Virginia, Charlottesville, VA 22904 USA. E-mail: stankovic@cs.virginia.edu.*
- *Tian He is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA. E-mail: tianhe@cs.umn.edu.*
- *George J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA. E-mail: pappasg@seas.upenn.edu.*

Manuscript received 24 Aug. 2020; revised 2 June 2021; accepted 15 June 2021. Date of publication 22 June 2021; date of current version 6 Jan. 2023. (Corresponding author: Yukun Yuan.)
Digital Object Identifier no. 10.1109/TMC.2021.3091324

1536-1233 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

subway stations [2], [3]. A few recent works have proposed solutions for subway system disruptions, including robust train schedules [4], timetable adjustment [5], taxi recovery services [6], and simple integration between bus and subway systems [7]. However, these works employ localized solutions with static routes or fixed schedules, without dynamic coordination of multiple transport modes.

To achieve sufficient resilience under disruptive events, it is critical to optimally control and coordinate all transport modes according to real-time and predicted demand with a global view. In this paper, we design a receding horizon control based dynamic integration framework called eRoute, which dynamically selects heterogeneous urban transportation systems, e.g., taxis, buses, and trains, and coordinates them to match passenger mobility patterns under specific disruptions. To conduct efficient management, eRoute considers several mobility-related factors, e.g., locations of disruptions, real-time information (e.g., location and occupancy status) from vehicles, the number of accumulated passengers, and the model of future passengers' mobility. The information of these factors is extracted from multi-source data of mobile devices and passengers, provided by existing sensing infrastructures, e.g., smartcard reader systems, and GPS devices in vehicles.

The eRoute framework features a two-level selection and coordination algorithm of heterogeneous transportation systems for disruptions, considering multiple factors, including both disruption factors, e.g., locations, scale, and urgency of disruptions, and transportation system factors, e.g., availability, cost, and efficiency of each transportation system. At the first level, parts of the subway and bus systems are selected, and then dynamically coordinated to help stranded passengers by city authorities. We formulate an optimal control problem that aims at maximizing the number of served passengers and minimizing the cost regarding extra traveling time and the number of extra vehicles. If the subway-bus integrated network cannot provide sufficient transport capacity to move stranded passengers, the second level is to engage the taxi companies and actively dispatch nearby available taxis to bridge impacted stations and surrounding bus and subway lines. Although taxis provide more flexible services in spatiotemporal dimensions than the bus and subway, they can be costly and also introduce local congestion, so it is used as the second-level solution.

The contributions of this work are listed as follows.

- To our knowledge, we conduct the first study on how to address disruptions in urban mobile systems based on real-time multi-source mobility data. We take the disruptions to urban transportation systems as an example, but our design can be generalized to a broader range of urban mobile systems.
- We design a service framework called eRoute, which dynamically selects and coordinates heterogeneous mobile systems according to the characteristics of the disruptive events and the mobile systems. To handle disruptions in the context of urban transportation systems, we formulate a dynamic subway-bus-taxi selection and integration problem for the disconnected transportation network under disruptions as an integral multicommodity max-flow problem with uncertain edge capacity. We prove the NP-hardness of this problem. We also design a two-level coordination algorithm to serve stranded passengers. The first level of coordination is to dynamically integrate subway and bus that are directly controlled by city authorities; and the second level coordination is to engage taxi companies to dispatch available taxis if stranded passengers are not fully served by the other two transportation systems.
- To determine the solution of the subway-bus integration problem in the first level of the coordination algorithm, we design a hierarchical Receding Horizon Control framework to adapt our solutions according to both current and estimated future passenger demand. At the high end of the hierarchy, we maximize satisfied passenger demand by obtaining rerouting and reallocation decisions for the overloaded transportation systems, meanwhile, at the low end of the hierarchy, we choose one of the optimal solutions obtained at the high end that also minimizes the cost of rerouting and reallocation. In the second level of the coordination algorithm, we formulate a taxi assignment problem as an integer linear programming problem and obtain the assignment solution by solving an approximation problem.
- We implement and evaluate eRoute with our datasets that consist of a bus system with 13,000 buses, a subway system with 123 stations, and an automatic fare collection system with a total of 16,840 mobile and static readers capturing 228,000 subway and bus passengers per hour from the same city Shenzhen. Compared to existing approaches with real-time data-driven features, our solution improves the ratio of served passengers per time interval by up to 11.5 times and reduces the average traveling time by up to 82.1 percent.

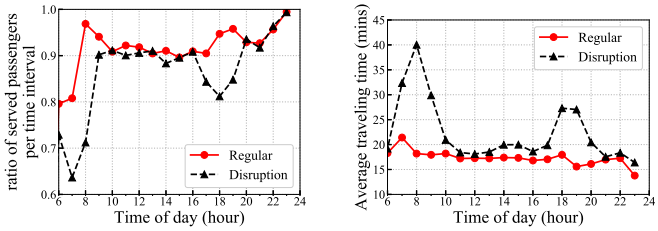
2 Motivation

2.1 Service Disruptions

Service disruptions of public transportation systems have significant impacts on passengers. They not only introduce travel delays, but also reshape mobility patterns, generating high operation costs due to longer travel distance, local congestion, and the resulting opportunity losses [2].

To fully understand various types of disruptions that occur in cities, [1] provides a taxonomy of disruptions in subway and bus systems. To summarize, there are multiple causes of transportation system failure, e.g., signal problems, weather disasters, power failures, construction, and mechanical failures, etc. These causes can result in reduced train or vehicle speed, service cancellation, and delay. Some of these disruptions affecting a small number of passengers can be handled within a single transportation service. For example, when a bus has a mechanical failure, another bus can be rescheduled to replace it. However, some other disruptions, e.g., shutdown of a subway station, can affect many passengers and require cooperated responses from multiple transport modes. This paper aims to address this type of significant disruption.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.



(a) ratio of served passengers per time interval of a day (b) Average traveling time

Fig. 1. Transportation system performances under disruptions.

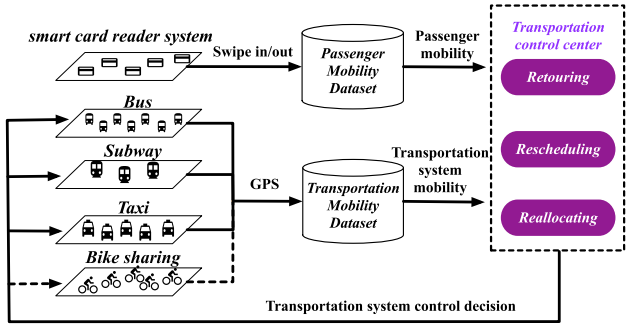


Fig. 2. eRoute framework.

We investigate an incident that occurred in Shenzhen, a metropolitan in China with our datasets. A signal problem caused long delays for all trains of a major subway line during the day. Both speed and frequency of trains along this line were significantly reduced due to safety concerns. No bus shuttles were used to address this incident.

We use the smart card reader dataset from the city to analyze the service quality of the subway on a day without disruption and a day with disruption. There are two measurement metrics, i.e., the ratio of served passengers per time interval and the average traveling time. Since the disruptive event happened on a weekday in the dataset, a weekday without disruptions in the same week is chosen as the regular day. Here, the ratio of served passengers per time interval represents the ratio between the number of actually served passengers during a time interval and the number of accumulated passengers who need service during the time interval, which is obtained by historical data assuming a stable daily passenger demand. In Fig. 1a, two curves show the ratio of served passengers per time interval (RSPI) on a regular day and a day with the disruption. We observe that the RSPI during peak-hours (6 am $\sim$ 9 am and 5 pm $\sim$ 8 pm) with the disruption is 35 percent less than that on a regular day. Meanwhile, as shown in Fig. 1b, the average traveling time of passengers with the disruption is 31 percent higher than that on a regular day. The average traveling time increased from 18 minutes to 40 minutes in the morning rush hours. These figures indicate that existing solutions do not effectively handle such disruptions.

2.2 eRoute Framework

To address one or multiple simultaneous disruptive events to urban transportation systems, we design a service framework called eRoute, which is a system that can be used by transportation authorities during disruptive events. Fig. 2 shows an overview of eRoute. Our main goal is to balance the supply and demand by dynamically integrating multiple urban mobile systems, e.g., subway, bus, taxi, and bike-sharing systems, etc. In this paper, we focus on three specific systems, namely subway, bus, and taxi. Given these systems, disruptive events usually change the topology of transportation networks, resulting in dramatically reduced supply at certain locations, and our solution is to automatically integrate heterogeneous mobile systems to increase supply for meeting passenger demand.

Typically, the supply of subway, bus, and taxi systems is organized in lines: (i) each subway or bus line has a fixed number of allocated vehicles; (ii) each subway or bus line has a route in which a few bus stops or subway stations are organized in a specific order; (iii) each subway or bus line has a schedule according to which the vehicles leave and arrive at certain bus stops or subway stations; (iv) one taxi is regarded as a line with flexible paths and pick-up time between stations and bus lines. With these features, the key idea of eRoute is to conduct the following three functions to deal with disruptive events:

- *Quantitative Reallocating*. Many disruptive events lead to failures of vehicles and a decrease of supply. Given the limited number of available vehicles for providing transport service, reallocating buses or trains among lines and assigning taxis to bridge influenced nodes and bus lines are critical to increasing supply with small overhead.
- *Spatial Rerouting*. This is an active traffic control strategy that presents alternate routes for buses, trains, and taxis. Rerouting is normally used when the regular route is severely affected by congestion and incidents. Here the purpose of rerouting is to re-balance supply with practical constraints across different regions under disruptive events. The alternate route information is disseminated to drivers using control channels in real-time.
- *Temporal Rescheduling*. Disruptive events directly affect the schedules of some transportation lines. eRoute reschedules the supply in other nearby lines and other transportation modes, increasing the current supply to the region to reinforce the service.

Different from existing works on transportation planning, eRoute is driven by real-time multi-source data, which has rich spatiotemporal information about passenger mobility patterns, and the supply and demand in transportation systems. Existing infrastructure in urban transportation systems already offers various data to the transportation center over the network in real-time. The smart card reader system records the events of every smartcard and then uploads them to the database at the transportation center. The format of the dataset is shown in Table 1. We extract past, current, and future passenger demand along every origin-destination (OD) pair from the dataset. The supply of every transport mode is obtained from GPS and occupancy datasets collected from every vehicle. For example, the GPS device in every bus reports the longitude, latitude, speed, and plate number of the bus, and the name of the bus line.

Under disruptions, the transportation network topology, and passenger demand change dynamically, so eRoute

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Dataset of Smartcard Reader

Collection period	2014/03/03~2014/03/09 2014/02/19~2014/02/20
Data size	10.4 GB
Record number	17,568,574
Format	[Device ID, Smartcard ID, Time, Metro station or Bus line] Bus boarding & Subway swiped-in/out

employs a two-level receding horizon control-based coordination framework to adapt control decisions based on both current and future passenger demand. In each iteration of the receding horizon control framework, eRoute first determines the first level coordination of subway and bus systems. Then in the second level coordination, if stranded passengers cannot be fully satisfied by the updated subway-bus integrated network, eRoute engages taxi companies to dispatch nearby available taxis to bridge stations and bus lines. Our framework selects part of urban transportation systems according to the features of disruptions and urban transportation systems.

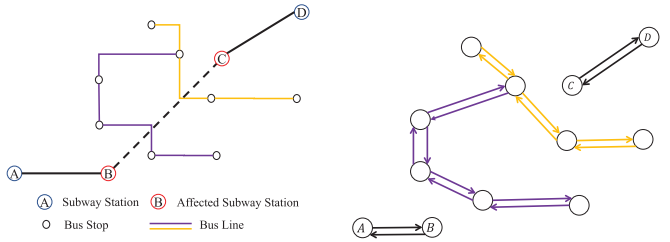
3 Integration of Subway and Bus

Both the subway system and bus system are the main components of the public transportation system, and they are complementary to each other. Either of them has a high potential to offload passengers from the other. In this section, we first study how to integrate subway and bus dynamically. Our design explores this potential to interconnect buses and subways dynamically to serve passengers under disruptive events. In this section, we formulate the one-iteration optimization problem of handling disruptions by controlling the subway-bus integrated network, including rerouting existing bus lines and reallocating extra buses and trains. Our goal is to provide alternative paths for stranded passengers that can meet a) dynamic passenger demand as much as we can with b) minimized cost including detour time due to rerouting and the number of extra vehicles needed. We show that our formulation is an integral multi-commodity maximum flow problem under dynamic edge capacity, and it is NP-hard.

3.1 Model Subway-Bus Integrated Network

Disruptions change the topology of urban transportation networks, and the transport capacity and cost between two stations. For example, if a subway or bus line cannot transport passengers between two stations of its route due to weather disasters or train mechanical failure, the corresponding edges in the transportation network should be removed. If a bus line is rerouted due to icy road conditions on its route, the capacity decreases and the cost increases for the corresponding edges.

Given a specific disruptive event, only part of subway and bus systems of the entire city are selected for integration to serve stranded passengers based on the features of disruptions, e.g., scale, urgency and location, and transportation system factors, e.g., availability, cost, and efficiency. For instance, if a disruptive event happens in a suburban area and there is no subway line nearby, only nearby bus lines are selected to transport stranded passengers. The large scale of influenced regions and a large number of stranded passengers mean that more subway lines and bus lines around the locations where disruptions happen should be selected.



(a) Subway-bus systems after a disruptive event (b) Subway-bus integrated network (G)

Fig. 3. Demonstration of the subway-bus integrated network after a disruptive event.

We use N^s and N^b to represent the numbers of selected subway lines and bus lines. We define the subway-bus integrated network after disruptions as $G = (V, E)$, where $V = V^s \cup V^b$. Every vertex in V^s denotes a subway station and every vertex in V^b represents a bus stop. Here we call either a subway station or bus stop as a node in the subway-bus integrated network. For any two vertices $v_i, v_j \in V$, if they are visited by the same subway or bus line consecutively, we add one directed edge, $e(v_i, v_j)$ from v_i to v_j. There are two attributes of every directed edge e_k: capacity and cost denoted as $w(e_k)$ and $p(e_k)$. Fig. 3 shows how to define the subway-bus integrated network after a disruptive event based on the subway-bus systems. In Fig. 3a, the subway connection between subway stations B and C breaks due to a disruptive event, and there are two bus lines around the two subway stations. Then the subway-bus integrated network after the disruption, G, is constructed as shown in Fig. 3b.

Suppose there are l nodes whose services are impacted due to disruptions and let $\mathbf{D} = \{v_{d_1}, v_{d_2}, .., v_{d_l}\}$ be the set of impacted nodes. For example, the services of four subway stations are ceased because of power failure. Then the four corresponding nodes of the ceased subway stations are regarded as the affected nodes. Passengers at these impacted nodes need to find alternative ways to reach their destinations. We define the origin-destination (OD) pair as (s_i, t_i), representing that there are passengers traveling from v_{s_i} to v_{t_i}. We use the number of stranded passengers to reflect the magnitude of disruptions and let C_i denote the passenger demand of one OD pair (s_i, t_i) during one time slot, e.g., 20 mins. The OD pairs and corresponding demand indicate passenger mobility patterns, which are learned from historical data sets.

3.2 Subway-Bus Integration Problem Statement

Definition 1 (Dynamic integration problem (DIP)). *Given the remaining subway-bus integrated network after disruptions, the problem is how to reroute existing N^b bus lines, reallocate extra N_b buses and N_s trains, and reschedule them to maximize the number of passengers that the new subway-bus integrated network can carry under practical constraints, i.e.,*

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

the detour time of a bus line is upper bounded, and the number of trains that can be deployed along a subway line is bounded.

$X^r \in \{0,1\}^{N^b \times l}$ is the decision matrix for rerouting, where $X^r_{k,i} = 1$ if kth bus line is rerouted to ith impacted transportation node. $X^b \in \{0,1\}^{N_b \times N^b}$ is the decision matrix for reallocating extra N_b buses, where $X^b_{h,k} = 1$ if hth extra bus is reallocated to kth bus line, otherwise, it is 0. $X^s \in \{0,1\}^{N_s \times N^s}$ is the decision matrix for reallocating extra N_s trains, where $X^s_{h,k} = 1$ if hth extra train is reallocated to kth subway line.

For every OD pair, we define a passenger flow from a source node s_i to a sink node t_i in the network G: an $s_i - t_i$ flow is a function $f : E \to \mathbb{R}^+$ that assigns a real number to each edge. Intuitively, $f(e) \geq 0$ is the amount of flow carried on the edge e, which represents the number of passengers transported along the edge e. There are two constraints: (1) capacity constraint: $\forall e \in E, f(e) \leq w(e)$; (2) flow reservation on transit node: for each node v except s and t, we have $\sum_{e \text{ into } v} f(e) = \sum_{e \text{ leaving } v} f(e)$.

Let S_i be the number of passengers that the network G can carry for an OD pair (s_i, t_i), subjected to the link capacity constraint and S_i is called the supply for (s_i, t_i). When the passenger demand of an OD pair (s_i, t_i) is less or equal to S_i, the network G can fully transport all the passengers without delay. Under disruptive events, there could be multiple OD pairs that need to be addressed simultaneously, and the supplies for these OD pairs are usually significantly insufficient, so our goal is to maximize the supplies $J = \sum_i S_i$.

Then we have the following realistic constraints for *DIP*: (i) Detour Constraint: if $X^r_{k,i} = 1$, $f(k,i) \leq \alpha$, where $f(k,i)$ is the function of extra detour time due to rerouting kth bus line to v_i, and α is the upper bound threshold of such detour time. It ensures that the increase of traveling time for regular bus passengers is not too high. We will discuss how to calculate $f(k,i)$ later. (ii) Allocation Constraint: $X^b \mathbf{1}_{N^b} \preceq \mathbf{1}_{N_b}$ and $X^s \mathbf{1}_{N^s} \preceq \mathbf{1}_{N_s}$, since every extra bus or train should be reallocated to at most one line, where $\mathbf{1}_{N_b}$ is a column vector of all 1s and the length of this vector is N_b. (iii) Schedule Constraint: $(X^s)^T \mathbf{1}_{N_s} \preceq \beta$, where β is a length N^s column vector and N^s is the number of subway lines considered in our problem. Let β_k denote the number of extra trains which can be reallocated to kth subway line for $1 \leq k \leq N^s$. We define this constraint, since there exists limitation of the number of trains operated along the same route for safety. (iv) Supply Constraint: To keep high utilization of our limited resource, we constrain that $\forall (s_i, t_i),\ S_i \leq C_i$, which specifies that the supply of one OD pair is less than or equal to its demand.

3.3 Subway-Bus Integration Problem Analysis

There are some similarities and differences between our problem and the classical max flow problem: we use the source-sink pair to describe every OD pair in G. Considering the integrated network G, simultaneously moving passengers for every OD pair means finding feasible integral flows in G. Intuitively, the objective of *DIP* is maximizing the sum of the size of every source-sink pair's flow. However, compared to the classical integral multicommodity max-flow problem, *DIP* has the following differences. (i)

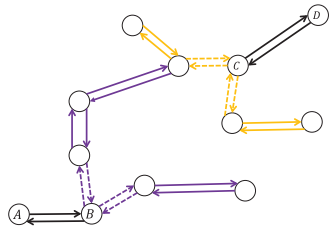


Fig. 4. Integrated network after rerouting (G').

Dynamic graph topology: in *DIP*, the network topology is dynamic because the rerouting decision X^r affects the edges in the network. For example, when one bus line is rerouted to pass a subway station d_i between two previously consecutive bus stops, two new edges are added to connect the bus line with the subway station, and the edge between these two previously consecutive bus stops is removed. (ii) Dynamic edge capacity: due to the reallocation of extra buses and trains, the capacity of some subway and bus lines would increase dynamically. Therefore, the capacity of corresponding edges in G also dynamically changes based on the reallocation decision, X^s and X^b. (iii) Constrained capacity of flow: due to the supply constraint stated previously: the supply of one OD pair is no more than the demand, the flow size of every source-sink pair should be no more than the corresponding demand.

To address dynamic graph topology and constrained capacity, we transform DIP to an Integral Multicommodity Max-Flow problem (IMCMF) under dynamic edge capacity by the following steps. The first step is to remove the dynamic graph topology by rerouting nearby bus lines to all the impacted nodes, if such rerouting decisions do not conflict with the detour cost constraint. Fig. 4 shows an example of the integrated network after applying the first step to network G (Fig. 3b). The second step is to add corresponding virtual source and sink nodes, and virtual edges to connect them with the impacted nodes. The capacity of a virtual edge represents the amount of passengers traveling between the corresponding origin-destination pair. Fig. 5 shows the integrated network after applying the second step to the network G'. By the second step, two virtual sink nodes (B'' and C''), two virtual source nodes (B' and C'), and the virtual edges are added to G'. The capacity of edge (B', B) and (C, C'') is equal to the passenger demand from B to C, and that of edge (B, B'') and (C', C) is equal to the passenger demand from C to B. In G'', the $B - C$ and $C - B$ flow are transformed to the $B' - C''$ and $C' - B''$ flow. Due to space limitation, the detail of these two steps is introduced in [1]. After the above two steps, *DIP* is changed to find the IMCMF with dynamic edge capacity in a subway-bus integrated network graph G''.

3.4 IMCMF Under Dynamic Edge Capacity

Let $X \in \mathbb{N}_0^{M \times N}$ denote the decision variable of the IMCMF, where M and N are the numbers of source-sink pairs and edges in G'' separately. $X_{i,j}$ represents the size of ith source-sink pair's flow along jth edge in G''. Then the objective is: $\max \sum_{i=1}^{M} \sum_{j=1}^{N} X_{i,j} R_{i,j}$, where $R \in \{0,1\}^{M \times N}$ is the relation matrix for calculating the size of flow of every source-sink pair. $R_{i,j} = 1$ if jth edge connects with the source node of ith pair. According to Step 2 in the previous section, only

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

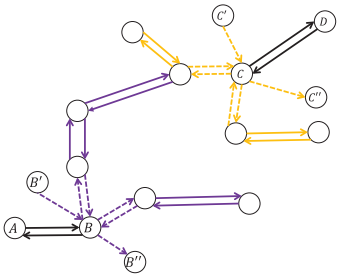


Fig. 5. Integrated network after adding virtual source and sink nodes (G'').

one edge is calculated for every OD pair, $\sum_{j=1}^{N} R_{i,j} = 1$ and $S_i = \sum_{j=1}^{N} X_{i,j} R_{i,j}$.

We define $R^{se} \in \{0,1\}^{N^s \times N}$ to represent the relation between subway lines and edges. $R^{se}_{j,k} = 1$, if kth edge is created due to jth subway line, otherwise, it is 0. $R^{be} \in \{0,1\}^{N^b \times N}$ denotes the relation between bus lines and edges. $R^{be}_{j,k} = 1$, if kth edge is created due to jth bus line, otherwise, it is 0.

Let $w^s(e_j)$ and $w^b(e_j)$ denote the capacity increase of edge e_j due to reallocating extra trains and buses respectively. We have the following equations:

$$w^s(e_j) = \sum_{i=1}^{N_s} \sum_{k=1}^{N^s} I(X^s_{i,k}) \times C^s \times R^{se}_{k,j}, \quad 1 \le j \le N \tag{1}$$

$$w^b(e_j) = \sum_{i=1}^{N_b} \sum_{k=1}^{N^b} I(X^b_{i,k}) \times C^b \times R^{be}_{k,j}, \quad 1 \le j \le N, \tag{2}$$

where the indicator function $I(X^s_{i,k}) = 1$ if and only if $X^s_{i,k} > 0$, otherwise, it is 0. C^s and C^b are the capacities of one train and bus respectively. Considering the edge capacity constraint, we have the constraint

$$\sum_{i=1}^{M} X_{i,j} \le w(e_j) + w^s(e_j) + w^b(e_j), \qquad 1 \le j \le N, \tag{3}$$

where $w(e_j)$ is the original capacity of edge e_j. We formulate the flow conservation on transit nodes: the amount of a flow entering an intermediate node is the same that exits the node. Therefore, for ith source-sink pair and kth node satisfies

$$\sum_{j=1}^{N} X_{i,j} R^{ne}_{k,j} = 0, \quad 1 \le i \le M, \quad 1 \le k \le L, \tag{4}$$

where $R^{ne} \in \{-1,0,1\}^{L \times N}$ describes the node-edge relation. L is the number of regular nodes in G''. $R^{ne}_{k,j} = 1$ if jth edge points to kth node and it is -1 if jth edge emits from kth node. Otherwise, it is 0. The IMCMF problem we consider is

$$\begin{aligned} \max \quad & \sum_{i=1}^{M} S_i = \sum_{i=1}^{M} \sum_{j=1}^{N} X_{i,j} R_{i,j} \\ \text{s.t.} \quad & S_i \le C_i, \quad X^b \mathbf{1}_{N^b} \preceq \mathbf{1}_{N_b}, \quad X^s \mathbf{1}_{N^s} \preceq \mathbf{1}_{N_s}, \\ & (X^s)^T \mathbf{1}_{N_s} \preceq \beta, \quad (1), \quad (2), \quad (3), \quad (4) \end{aligned} \tag{5}$$

Theorem 1. *The* DIP *is NP-hard.*

Proof. Compared with maximum integral multi-commodity flow, our problem has dynamic edge capacity and the edge capacity is also one decision variable. In *DIP*, let $N_b = N_s = 0$, then the maximum integral multi-commodity flow problem is one special case of *DIP*. [8] shows that maximum integral multi-commodity flow is NP-hard. Hence, *DIP* is NP-hard. □

Based on the literature [8], [9], the integral maximum multi-commodity flow problem is Max SNP-hard even in several particular cases. This result implies that there exists no polynomial-time approximation scheme unless P=NP. [9] proves that IMMF is not only strongly NP-hard but finding an approximate solution within a fixed performance ratio for it is still one NP-hard problem. [10] shows that it is NP-hard to approximate within $m^{\frac{1}{2}-\epsilon}$, where m is the number of edges. Although there exist several works providing one approximation algorithm or linear time algorithm, they require that the graph is one tree [8]. Meanwhile, our problem is still different from the existing dynamic graph problem, where edge capacity changes with time, but it is not one decision variable [11].

Although the *DIP* is hard to solve in nature, we argue that the disruptions in urban transportation systems usually only affect small numbers of stations and the subway lines and bus lines around them. Therefore, the input size of the DIP is not large. Based on our linear integer programming (LIP) formulation of the problem, existing solvers can solve them relatively quickly. In our evaluation, the optimal solution to the problem can be obtained within one minute, which is fast enough for transportation system control in reality.

4 Design of RHC Based Subway-Bus Integration Algorithm

The transportation control center receives real-time streaming data including smart card records, vehicles' GPS locations, and occupancy status periodically. These real-time data streams are then processed to predict the spatiotemporal patterns of passenger demand. Based on the prediction, the control center utilizes a receding horizon control (RHC) algorithm to calculate a control solution periodically in real-time, to match predicted passenger demands.

To obtain optimal bus/train line rerouting and extra bus/train reallocation decisions, we consider two main objectives: 1) maximize the passenger transport for the overload transportation systems under disruptive events; 2) minimize the rerouting and reallocation cost while achieving the maximum passenger transport. When disruptive events happen, the eRoute system should adapt the solutions to both current and possible future demand. Hence, we design a hierarchical RHC algorithm. The high-level problem is based on our problem formulation in Section 3, which suggests the passenger flows and their paths (rerouting decisions), and the reallocation of buses and trains. The low-level problem is to minimize the rerouting cost and reallocation cost regarding extra detour time and the number of additional buses or trains reallocated.

4.1 Variables, Constraints and Objective Functions

We assume that the optimization time horizon is T, indexed by $t = 1, \ldots, T$. We first reformulate the variables in the

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

optimization time horizon. Let $w(e_k, t)$ be the capacity of e_k during the time slot t. $C_i(t)$ represents the passengers demand of ith source-sink pair (s_i, t_i) during time slot t, which can be predicted based on the historical dataset and real-time sensor information. We define $X_{i,j}(t) \in \mathbb{N}_0^{M\times N}$ as the decision variable of the integral multicommodity max-flow during time slot t. Meanwhile, $X^b(t) \in \{0,1\}^{N_b\times N^b}$ and $X^s(t) \in \{0,1\}^{N_s\times N^s}$ represent the reallocation decision of extra buses and trains during time slot t.

Modeling Circulating Bus Supply. The routing and allocation in subway-bus networks have to meet spatiotemporal constraints, due to operating schedules and road conditions. For example, a bus may become available for reallocation after transporting all passengers at its final stop in a finite optimization horizon. We define $W^b \in \mathbb{N}_+^{N^b}$, one column vector to denote the number of time slots needed to complete one end-to-end trip of all bus lines. For instance, W_i^b is the number of time slots needed to finish one trip of ith bus line. Let $U^b(t) \in \mathbb{N}_+^{N_b}$ be one column vector to represent the number of time slot needed to finish current bus line trip at the beginning of time slot t. We note that $U^b(t)$ may change over time due to congestion and road conditions. More importantly, it is directly affected by the rerouting decision. For instance, it costs 4 time slots to finish one trip of the first bus line, and the first extra bus reallocated to the first bus line at time slot 1. So at the time slot 2, we have the following values: $U_1^b(2) = 3$. Then the relation between $U_i^b(t)$ and $U_i^b(t-1)$ is

$$U_i^b(t) = \max\left\{0, \max\left\{\sum_{j=1}^{N^b} X_{i,j}^b(t-1)W_i^b, U_i^b(t-1)\right\} - 1\right\}, \tag{6}$$

where $t \geq 2$ and $U_i^b(1) = 0$ for $1 \leq i \leq N_b$. Based on $U^b(t)$, $\gamma^b(t) \in \{0,1\}^{N_b}$ is one vector column to describe whether every extra bus can be reallocated during time slot t. It is clear that if $U_i^b(t) > 0$, ith extra bus is still operating for one existing bus line and it cannot be reallocated, otherwise, it can be reallocated. We have the following equation:

$$\gamma_i^b(t) = I_1(U_i^b(t)), \tag{7}$$

where $I_1(U_i^b(t))$ is an indicator function, and it is equal to 1 if $U_i^b(t) = 0$, otherwise, it is 0. Then for ith bus, during time slot t, it cannot be reallocated to more than $\gamma_i^b(t)$ bus lines

$$X^b(t)\mathbf{1}_{N^b} \preceq \gamma^b(t). \tag{8}$$

We remark that it's possible that $\sum_{j=1}^{N^b} X_{i,j}^b(t) = 0$, however, ith bus also contributes to one existing bus line, because of operating for one existing bus line. Hence, we define $O^b(t) \in \{0,1\}^{N_b\times N^b}$ to denote which bus line that every extra bus contributes to during the time slot t. $O_{i,j}^b(t) = 1$ if ith bus is operated for jth bus line during time slot t, otherwise, it is 0. Then, we have the following relation:

$$O_{i\cdot}^b(t) = O_{i\cdot}^b(t-1)I_2(U_i^b(t)) + X_{i\cdot}^b(t), \tag{9}$$

where $O_{i\cdot}^b$ is the ith row of $O^b(t)$ and $I_2(U_i^b(t))$ is also one indicator function. $I_2(U_i^b(t)) = 1$ if $U_i^b(t) > 0$, otherwise, it is 0. Finally, we describe the capacity increase of e_j during time slot t due to N_b extra buses

$$w^b(e_j, t) = \sum_{i=1}^{N_b}\sum_{k=1}^{N^b} I(O_{i,k}^b(t)) \times C^b \times R_{k,j}^{be}, \tag{10}$$

where C^b is the capacity that one extra can provide. The circulating supply model and constraint of subway trains are similar to that of buses.

Modeling Circulating Train Supply. The model for circulating train supply mirrors the one for "circulating bus supply". There is only a symbol (s instead of b) that is replaced and everything else carries through. To avoid redundancy, we only show the equations for this model as follows and the notations have a similar definition to that in the model of circulating bus supply

$$U_i^s(t) = \max\left\{0, \max\left\{\sum_{j=1}^{N^s} X_{i,j}^s(t-1)W_i^s, U_i^s(t-1)\right\} - 1\right\} \tag{11}$$

$$\gamma_i^s(t) = I_1(U_i^s(t)) \tag{12}$$

$$X^s(t)\mathbf{1}_{N^s} \preceq \gamma^s(t) \tag{13}$$

$$O_{i\cdot}^s(t) = O_{i\cdot}^s(t-1)I_2(U_i^s(t)) + X_{i\cdot}^s(t) \tag{14}$$

$$w^s(e_j, t) = \sum_{i=1}^{N_s}\sum_{k=1}^{N^s} I(O_{i,k}^s(t)) \times C^s \times R_{k,j}^{se} \tag{15}$$

$$\sum_{i}^{M} X_{i,j}(t) \leq w(e_j, t) + w^s(e_j, t) + w^b(e_j, t), \tag{16}$$

$$\sum_{j=1}^{N} X_{i,j}(t)R_{k,j}^{ne} = 0,\ 1 \leq i \leq M, 1 \leq k \leq L. \tag{17}$$

4.2 A Hierarchical RHC Algorithm

Our goal of the high level RHC problem formulation is to seek the dynamic rerouting and reallocation decision based on predicted passenger demand.

This formulation is based on the problem transformation in the previous section

$$\begin{aligned} \max_{X(t),X^b(t),X^s(t)} \quad & \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}(t)R_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{N} X_{i,j}(t)R_{i,j} \leq C_i(t),\ (X^s)^T\mathbf{1}_{N_s} \preceq \beta,\ (6) \sim (17). \end{aligned} \tag{18}$$

After solving the above problem, we obtain the maximum demand that the system can support, which is equal to the amount of the supply that the system needs to provide.

In general, the problem (18) has multiple optimal solutions, i.e., different flow and reallocation assignments can

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

achieve the same supply in the integrated network. Assume the optimal solutions of (18) is a set $\{\hat{X}(t), \hat{X}^s(t), \hat{X}^b(t)\}$. They provide the maximum value of supply. Therefore, we introduce the low-level problem which is formulated to choose the optimal flow and reallocation assignments with the minimum cost. At this level, the goal is rerouting existing bus lines and reallocating the extra vehicle supplies along different lines with minimal cost. We minimize the cost to satisfy the supply achieved in (18), which consists of the rerouting cost and the number of extra buses or trains.

Algorithm 1. RHC Algorithm for Real-Time Subway and Bus System Control

Input: Time horizon T minutes, period of updating solution t_1 minutes; number of time slots to finish one bus or subway line trip W^b, W^s; train and bus capacity C^s, C^b; geometrical information of transportation nodes; historical and real-time data of smart card events; real-time vehicle trajectory data; parameter θ.
Output: Control decision: X^r, X^s, X^b
while At the beginning of every t_1 minutes **do**
Update the number of available buses, N_b and trains, N_s; update the passenger demand of every OD pair; update prediction of $C_i(t)$ for the time horizon T; update the edge capacity during the time horizon T, $w(e_j, t)$; update the parameter β
Solve the max-flow problem (18) to get the optimal solution set $\{\hat{X}(t), \hat{X}^s(t), \hat{X}^b(t)\}$ of problem (18)
Solve the min-cost problem (19) to get the control decision with the minimum cost.
Send the control decisions according to solution: X, X^s, X^b.
end while
return Control decision

Rerouting Cost. it is defined as: $J_r = \sum_{j=1}^{N} I(\sum_{i=1}^{M} X_{i,j}(t))p(e_j)$, where $p(e_j)$ is the cost of traveling along edge e_j.

Cost of Reallocation. it is defined as the number of extra buses and trains used, denoted by: $J_n = \sum_{k=1}^{N_b} \sum_{j=1}^{N^b} X^b_{i,j}(t) + \sum_{k=1}^{N_s} \sum_{j=1}^{N^s} X^s_{i,j}(t)$.

We define a weight parameter θ when summing up the costs related to both objectives. The formulation of minimizing cost is shown as follows:

$$\min_{X(t),X^b(t),X^s(t)\in\{\hat{X}(t),\hat{X}^b(t),\hat{X}^s(t)\}} J = \sum_{t=1}^{T}(J_r + \theta J_n). \tag{19}$$

4.3 RHC Framework Implementation

We adopt a basic linear regression technique to predict passenger demand of different OD pairs based on historical and real-time datasets. We define the time horizon as T minutes and the length of every time slot is t_1 minutes. The previous proposed RHC based problem formulation is embedded in one iteration of our RHC algorithm, and we update the control decision every time slot, t_1 minutes. The pseudo-code of the RHC algorithm is shown as Algorithm 1. For simplicity, we assume that the two-level RHCs have the same timescale.

This RHC algorithm is triggered when one or multiple disruptive events occur in the transportation systems, which cause subway stations or bus stops to close. This algorithm periodically makes control decisions every t_1 minute until the transportation system recovers from the disruption. At the beginning of every t_1 minute, it updates the locations and occupancy status of all the available extra buses and trains and predicts passenger demand of every OD pair till the future T time horizon. Then it solves the problem (18) to the optimal solution set to transport the maximum number of passengers, solves the problem (19) to obtain the rerouting decision of existing bus lines, extra bus and train assignments that minimizes the control cost.

5 Heterogeneous Transportation Modes Integration

Although rerouted bus lines and reallocated buses and trains can transport stranded passengers, there may still be some passengers that cannot be picked up under the subway-bus integrated network since (i) there are not enough bus lines around the impacted nodes and (ii) there exists competition between regular bus passengers and stranded passengers due to limited transport capacity.

Hence, if the subway-bus integrated network after rerouting and reallocation cannot satisfy all stranded passengers, taxi companies are engaged to dispatch nearby available taxis for increasing transportation system capacity. The reason for integrating heterogeneous transportation modes with priority, (i.e., subway and bus systems have the priority, and taxis have the second priority) is that both subway and bus are public transportation modes that can be controlled by urban transportation authority, whereas taxis belong to private companies. Meanwhile, taxi service is costly and also introduces local congestion.

5.1 Model Taxi Dispatch Decisions

According to the factors of both disruptions and urban transportation systems, we select the available taxis which are close to locations where disruption happens, or close to the previously selected subway and bus lines for dispatching. Engaging taxi service takes advantage of its flexibility to reduce passenger congestion in impacted nodes where there is not enough subway and bus system supply. Taxis are used to connect impacted nodes and bus lines (node-line pairs) to transport as many passengers as possible, meaning taxis are dispatched to node-line pairs.

Given passenger demand in the impacted nodes and the subway-bus integrated network with extra virtual bus edges, we define the node-line pairs which need to be served by extra taxis as follows. Given one impacted node, we first decide whether this node needs the taxi service. In the subway-bus integrated network with virtual bus edges, we calculate the pickup and drop-off capacity provided by the bus network and then compare such capacity with leaving and arriving passenger demand in the node. If pickup capacity or drop-off capacity is smaller than leaving or arriving passenger demand, this node is selected to be served by taxis. For one impacted node selected in the previous step, we determine a group of bus lines that are linked to the node by taxis. These bus lines are selected if the bus

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

line can be reached from the node within one time slot, and the line does not pass by the node after adding virtual bus edges. It is noted that node-line pairs are selected according to passenger demand and subway-bus integrated network transport capacity. Due to dynamic passenger demand and bus network supply, the group of selected node-line pairs changes over time.

We assume that there are total L node-line pairs, meaning there are L dispatch options for one taxi at one time slot. Suppose there are N_t taxis available for dispatching, which are selected based on their locations and occupancy status. At the beginning of each time slot, the transportation control center collects the real-time locations and occupancy status of taxis. Only the unoccupied taxis that are able to reach the impacted stations or bus lines within a time slot are recruited for taxi dispatch. We use N_t to represent the number of these taxis. Let $X^t(t) \in \{0,1\}^{N_t \times L}$ be the decision matrix for dispatching N_t taxis, where if $X^t_{i,j}(t) = 1$, ith taxi is dispatched to jth node-line pair, otherwise, it is 0. The capacity of one node-line pair, e_j, is determined by taxi dispatch decisions, $w(e_j) = \sum_{i=1}^{N_t} X^t_{i,j}(t)$. We assume that the trip of one node-line pair can be finished within one time slot according to how we select these node-line pairs. We ensure that one taxi can be dispatched to at most one node-line pair during each time slot, defined as

$$\sum_{j=1}^{L} X^t_{i,j}(t) \leq 1, \quad 1 \leq t \leq T, \quad 1 \leq i \leq N_t. \tag{20}$$

5.2 Taxi Dispatch With Pre-Computed Solutions

In this subsection, we discuss how to dispatch taxis with the pre-computed rerouting and reallocation solutions. We first define the updated transportation network, given rerouting and reallocation decisions. Then we model taxi mobility patterns during our optimization horizon and discuss the constraints and objectives of taxi dispatch.

Updating Integrated Transportation Network With Rerouting and Reallocation Decisions. The integrated transportation network varies with time due to dynamic rerouting and reallocation decisions. We describe how to generate the transportation network $G_1(t)$ if applying rerouting and reallocation decisions to the previous remaining network. First, for any virtual bus edge e_j in the remaining transportation network that is generated by the problem transformation in Section 3.3, if there exists no flow through e_j and its virtual pair edge at time slot t, both two virtual bus edges will be deleted. Then the edge capacity of any edge e_j is equal to $w(e_j, t) + w^s(e_j, t) + w^b(e_j, t)$. Recall that if one bus line is rerouted to one impacted node, there are two virtual edges added, which are called the virtual pair edge of each other.

Let $\bar{X}(t) \in \mathbb{N}_0^{M \times \bar{N}}$ be the integer multicommodity maximum flow in $G_1(t)$ with dynamic taxi dispatch decisions $X^t(t)$, where $\bar{N}$ is the number of edges in $G_1(t)$.

Model Taxi Mobility Pattern. Given the origin and destination of L virtual taxi edges, we use $OP \in \mathbb{R}^{L \times 2}$ and $DP \in \mathbb{R}^{L \times 2}$ to denote the GPS position of the origin and destination of each virtual edge, respectively. Let $P_i(t) \in \mathbb{R}^{1 \times 2}$ be the estimated GPS information of ith taxi's location by the end of time slot t. Then if ith taxi is dispatched at time slot t, its end position $P_i(t)$ is $X^t_{i,\cdot} DP \in \mathbb{R}^{1 \times 2}$. If there is no dispatch decision for ith taxi, the taxi will drive on the road to find the next taxi passengers. We assume that the ending position for taxi i by the end of time slot t is related to the ending position at the end of slot $t-1$. The relation between $P_i(t)$ and $P_i(t-1)$ is formulated as

$$P_i(t) = f(P_i(t-1)), \quad f : \mathbb{R}^{1 \times 2} \rightarrow \mathbb{R}^{1 \times 2},$$

where the function f represents the data-driven taxi passenger mobility pattern. For example, this pattern is described by a matrix, where each element denotes the probability that a taxi drops off a passenger in a road segment at the end of slot $t+1$ when the trace starts from the position at the beginning of slot t. This pattern can be learned from the historical passenger trip data by some methods, e.g., [12], [13], and it approximately describes the passenger mobility pattern in the real-world [12].

To summarize, the ending position of ith taxi by the end of time slot t is defined as

$$P_i(t) = \left(1 - \sum_{j=1}^{L} X^t_{i,j}(t)\right) f(P_i(t-1)) + \left(\sum_{j=1}^{L} X^t_{i,j}(t)\right) X^t_{i,\cdot} DP. \tag{21}$$

Given the decision $X^t_{i,\cdot}$, if $\sum_{j=1}^{L} X^t_{i,j} = 1$, ith taxi is sent to a target location $X^t_{i,\cdot} OP$ at the beginning of time slot $t-1$ from position $P_i(t-1)$, the approximated idle driving distance is

$$d_i(t) = ||X^t_{i,\cdot} OP - P_i(t-1)||_1, \ i \in [1,, N_t], \ t \in [1, T]. \tag{22}$$

Here, to estimate the driving distance without knowing the exact path between two locations, we use the Manhattan norm or one norm between two geometry positions, which has already been used to model the trip distance in metropolitans like Beijing and Seoul [13], [14], [15], [16].

Constraints. Due to limited traveling speed and time, the traveling distance within one time interval should be bounded. For example, if the destination of virtual taxi edge e_{j_1} is far away from the origin of another virtual taxi edge e_{j_2}, one taxi cannot reach the origin node of e_{j_2} within one time slot. It means a taxi should not be dispatched to e_{j_1} and e_{j_2} at two consecutive time slots. The upper bound for taxi i is denoted by $\gamma_i \in \mathbb{R}$ and any $d_i(t)$ should satisfy

$$\sum_{j=1}^{L} X^t_{i,j}(t) d_i(t) \leq \gamma_i. \tag{23}$$

Considering the flow conservation on transit nodes, we define

$$\sum_{j=1}^{\bar{N}} \bar{X}_{i,j}(t) \hat{R}^{ne}_{k,j} = 0, \ 1 \leq i \leq M, 1 \leq k \leq L. \tag{24}$$

The edge capacity constraint of flow during every time slot $1 \leq t \leq T$ is defined as

$$\sum_{i}^{M} \bar{X}_{i,j}(t) \leq \bar{w}(e_j, t) + \bar{w}^t(e_j, t), \quad 1 \leq j \leq \bar{N}, \tag{25}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

where $\bar{w}^t(e_j,t) = C^t \times \sum_{i=1}^{N_t} X_{i,j}^t(t)$ and C^t is the capacity of each taxi.

Objectives. Our goal is to seek dynamic taxi dispatch, meanwhile, considering the number of stranded passengers that we can serve, the number of taxi passengers that may be missed, and the taxi dispatch cost regarded as idle driving distance. The first one is the primary objective of taxi dispatch to pick up the stranded passengers. Since dispatching taxis to pick up impacted subway or bus passengers may miss some taxi passengers, we consider reducing the number of missed taxi passengers as the second objective. Traveling from position $P_i(t-1)$ to $X_{i,\cdot}^t OP$ for service will introduce cost since taxi drives on the road without serving any passengers. Hence, we want to minimize this kind of idle driving distance while dispatching taxis.

Number of Passengers That Can be Served. According to our decision variables $\bar{X}(t)$ and $X^t(t)$, the number of stranded passengers that can be served by the subway-bus-taxi integrated network is $\sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N}\bar{X}_{i,j}(t)\bar{R}_{i,j}$, where $\bar{R} \in \{0,1\}^{M\times\bar{N}}$ is the relation matrix for calculating the flow size of each source-sink pair. $\bar{R}_{i,j}=1$ if jth edge is associated with the source of ith pair, otherwise, it is 0.

Number of Missed Taxi Passengers. $J_{tp}(t)$ is the number of missed taxi passengers at time slot t. Let $C^t(t)$ how many passengers one taxi serves during a time slot t that can be learned from the historical dataset. Then it is defined as $J_{tp}(t) = C^t(t)\sum_{i=1}^{N_t}\sum_{j=1}^{L} X_{i,j}^t(t)$.

Idle Driving Distance. According to the definition of $d_i(t)$, we formulate the total idle driving distance due to taxi dispatch at time slot t as: $J_t(t) = \sum_{i=1}^{N_t}\sum_{j=1}^{L} X_{i,j}^t(t)d_i(t)$.

In our design, we aim to co-optimize the above three objectives. The first two objectives are conflicting. For example, to serve more stranded passengers, more taxis should be recruited, thus reducing the taxi supply for taxi passengers. The enhanced eRoute balances these two objectives by using a weight parameter, i.e., δ_1 to determine how much effort to spend on assisting the stranded passengers. The last objective is a common cost for taxi dispatch, and we use the second parameter, i.e., δ_2 to balance it with the other two objectives. To summarize, we formulate the taxi dispatch problem as

$$\begin{aligned}&\max_{\bar{X}(t),X^t(t)} \quad \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{\bar{N}}\bar{X}_{i,j}(t)\bar{R}_{i,j} + \delta_1\sum_{t=1}^{T}J_{tp}(t) + \delta_2\sum_{t=1}^{T}J_t(t)\\ &\mathbf{s.t.} \quad \sum_{j=1}^{\bar{N}}\bar{X}_{i,j}(t)\bar{R}_{i,j} \le \bar{C}_i(t),\ (20)\sim(25).\end{aligned} \tag{26}$$

By solving this problem, the taxi dispatch decisions for N_t vacant taxis during the future T time slots is generated. Problem (26) is a mixed integer linear programming problem, which is not efficient to be solved regarding the problem size, i.e., the number of decision variables. We can relax the problem by replacing constraint $X_{i,j}^t(t)\in\{0,1\}$ with $0 \le X_{i,j}^t(t) \le 1$. With such approximation, the relaxed form of Problem (26) is a linear programming problem. According to the optimal solution $\hat{X}^t(t)$ of the relaxed form of Problem (26), we set $X_{i,j}^t(t)=1$ if $j = \arg\max_{j'}\{\hat{X}_{i,j'}^t(t)\}$, otherwise, it is 0.

5.3 RHC Based Coordination Algorithm of Heterogeneous Transportation Systems

The pseudo-code of the RHC based coordination algorithm is shown in Algorithm 2. The algorithm consists of two levels for computing the control decisions at the beginning of each time slot. In the first level, we obtain the decisions of integrating subway and bus systems by applying Algorithm 1 on the integrated network and then update the integrated transportation network after applying the decisions. If not all passengers can be served by the updated integrated transportation network, in the second level, we solve the problem (26) to determine the dispatch decisions of nearby sparse taxis. After these two levels, the control decisions of heterogeneous transportation systems are obtained.

Algorithm 2. RHC Based Coordination Algorithm of Heterogeneous Transportation Systems

Input: Time horizon T minutes, period of updating control solution t_1 minutes; taxi capacity C^t; geometrical information of taxis; parameter δ_1, δ_2
Output: Control decision: X^t

```
while At the beginning of every update period t_1 minutes do
    Run Algorithm 1 to determine decisions of subway and bus.
    if some passengers are not served by the transportation system with rerouting and reallocation then
        Generate dynamic integration network G_1 with previously calculated decisions.
        Update the locations and status of taxis.
        Solve the problem (26) to determine taxi dispatch decisions X^t using network G_1.
    else
        No control decisions for nearby sparse taxis.
    end if
end while
return Control decision of subway, bus, and taxi
```

This two-level integration algorithm dynamically selects whether the taxi system is engaged to serve passengers based on the first-level decisions of subway and bus systems and the predicted number of stranded passengers. Part of subway and bus systems are chosen for integration according to factors of disruption, (e.g., location), and mobile systems (e.g., availability).

6 Evaluation

6.1 Methodology

To evaluate eRoute in a real-world scenario, we use the dataset described in Table 1 to conduct a data-driven analysis. We can see that a smartcard record contains the location, time, and transport mode that one passenger swipes the smart card. Based on this dataset, we can extract the origin and destination of a trip for each passenger. Fig. 6 shows the passenger demand density of the subway and bus system over one city. Then we can predict the passenger demand of each OD pair using linear regression.

Given a day without disruptions and a day with disruptions, we count the passenger demand for each OD pair during each time slot. Then we compute the passenger

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

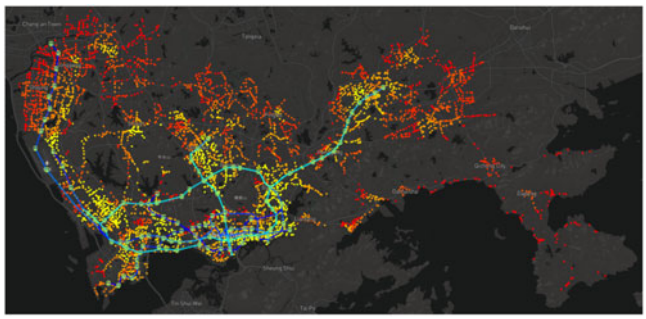


Fig. 6. Passengers demand density over the city (the lighter the icon, the higher the demand density).

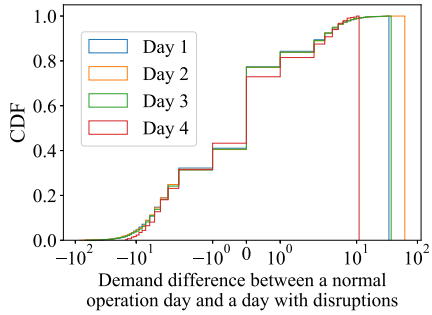


Fig. 7. Passenger demand change between a weekday with and without disruption.

demand difference for the two days of the same time slot and OD pair and plot the CDF figure in Fig. 7. We show the demand difference between each of four normal operation days and a day with disruptions. It is observed that the passenger demand changes little after the disruptive event. The reason is that the influenced subway line connects the central business area and the residential areas of the city. Although the subway line reduced the speed of trains under the disruptive event, the affected commuters still had to select this subway line due to the limited low-cost alternative transportation options. Based on this observation, in the evaluation, it is reasonable to infer the passenger demand on a day with disruptions using the collected passenger mobility data on normal operation days. The other disruptions that are not collected in the dataset, such as train crashes, may decrease the passenger demand on the corresponding transportation systems. However, the daily commuters usually have fixed origins, destinations, and departure time of their trips, which do not change after disruptions. They may travel with different transportation modes due to disruptions. It still makes sense to estimate how many passengers want to travel starting from or ending at an affected region of a city during each time slot after disruptions by the collected data of passengers during normal operation days. eRoute can use the estimation to integrate heterogeneous transportation modes, such as buses, taxis, and trains for serving affected passengers.

We also use a dataset of GPS traces of all buses along 800 different bus lines in the same city. Every bus has networked GPS that can upload real-time location information every 30 seconds. One record in this dataset contains a plate number, a bus line number, a timestamp in seconds, GPS Coordinates, and a real-time speed. Based on this dataset, we can estimate the schedules of every bus line and the trip time during the day. We can also estimate the real-time passenger demand and available capacity of one bus by combining smartcard reader data and bus GPS data, as all buses use the smartcard system. The locations of subway stations and bus stops are obtained from an online digital map service provider. The typical capacity of a city bus is 60 passengers. We assume 25 extra buses can be reallocated.

This dataset contains one disruptive event: a signal problem starting from 7 am to the end of the day that causes significant delays on all trains of a subway line. Both the train speed and the frequency of trains were reduced due to safety concerns. No bus shuttles had been used to address this incident in reality based on an analysis of our data.

To show the effectiveness of eRoute, we compare it with the following existing solutions to handle disruptions. (i) Periodic control with extra vehicles: this method discretizes the time of a day into multiple time slots and the length of a time slot is 20 minutes. Periodic control with extra vehicles updates the assignment of extra vehicles and rerouting decisions of existing bus lines at the beginning of each time slot. The objective of this method is to serve the current stranded passengers. (ii) Periodic control without extra vehicles: the timeline is discretized into multiple 20-minute time slots. This solution only reroutes the existing bus lines to help the stranded passengers. It updates the rerouting decisions at the beginning of each time slot. When calculating the rerouting decisions, the objective is to serve the current stranded passengers. (iii) Static: compared with eRoute, this method determines how to reroute the same existing bus lines and reallocate the same number of extra vehicles at the beginning of the disruptions. The rerouting and reallocating decisions do not change until the disruption is addressed. (iv) Shuttle: the transportation center utilizes the dedicated vehicles running along the influenced subway line to provide substitute services. The number of dedicated vehicles is equal to the number of extra vehicles used in eRoute. (v) Shuttle+Taxi: Besides the dedicated vehicles, a group of taxis (a hundred taxis) is also used to run along the influenced subway line and the capacity of each taxi is four. (vi) Nearest: for every influenced subway station, only ten geographically close bus lines are considered for rerouting and reallocation. This method also updates the decisions once per time slot. (vii) Enhanced eRoute: the transportation center reroutes bus lines and reallocates extra trains based on eRoute. If the influenced passengers are not fully served, some extra vacant taxis are dispatched based on Algorithm 2.

The performance metrics considered include: (i) the ratio of served passengers per time interval (RSPI): the number of passengers served during a time interval / (the number of accumulated unserved passengers at the beginning of the time interval + the number of new passengers during the time interval); (ii) average traveling time; (iii) response time: time to serve fixed percentage of passengers. In the experiment, for our eRoute, the length of every time slot is 20 minutes and then the time horizon is 6 time slots. In the evaluation, the optimal solution of rerouting, rescheduling, and reallocation can be obtained within two minutes using the Gurobi solver on a PC with an Intel i5 CPU and 8GB memory. All the vehicles start to execute the decisions after obtaining the decisions of dispatch.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

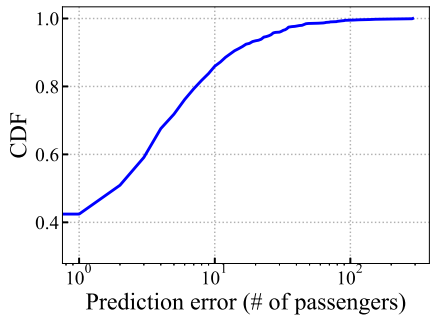


Fig. 8. CDF of prediction error.

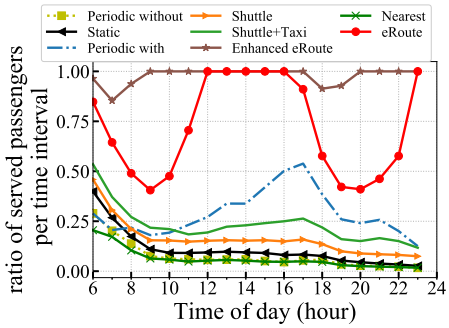


Fig. 10. Solutions' performance in unit time over a day.

6.2 Results

Prediction Error. In eRoute, we use one linear model to predict the passenger demand of different OD pairs during each time slot, then use them to run our RHC algorithm. We evaluate the accuracy of our prediction method by using one-day data as the testing set and five-day data as the training set. Figs. 8 and 9 show the CDF of prediction error and prediction error ratio respectively. Here, we have 72 OD pairs. 80.0 percent passenger demand of one OD pair during a time slot is no more than 30 and 50.0 percent of that is fewer than 10 passengers. The prediction error of nearly 90 percent of OD pairs is less than ten passengers, which demonstrates that a simple linear model can predict passenger demand fairly accurately. In Fig. 9, the prediction error ratio of some OD pairs is large since the actual passenger demand is very small (< 5) resulting in a large prediction error ratio.

Comparison of Solutions. Fig. 10 plots the ratio of served passengers per time interval of eight solutions over the day. The performance of eRoute decreases from 6 am to 9 am and then it increases with reduced passenger demand, but it still significantly outperforms the other solutions. Compared to the widely used Static solution, eRoute achieves up to 2.82 times higher RSPI during rush hours. This is because that eRoute benefits from our RHC by considering the passenger demand in the future several time slots. Even compared to the solution, i.e., periodic control with extra vehicles, eRoute achieves 64.1 percent higher RSPI during 19:00-19:59. In the off-peak hours, e.g., 12:00-15:59, with the decrease of passenger demand, eRoute can serve all the passengers. Enhanced eRoute can serve nearly all congested passengers except several peak hours, i.e., 7:00-8:59 and 18:00-19:59 compared with eRoute.

There are also several observations: the first one is updating the control decisions dynamically can improve the performance. Comparing static solution, eRoute and periodic control with extra vehicles, although all of the three solutions reallocate the same number of extra buses, the static solution has the lowest RSPI than the other two solutions do during most time of the day, such as only 50.0 percent of periodic control with extra vehicles and 20.1 percent of eRoute during 10:00-10:59. The second observation is that rerouting the existing bus lines can provide part of the substitute supply, but more extra buses are needed. The third one is that the transport capacity between influenced nodes and existing bus networks limits the total number of served passengers by comparing the performance of enhanced eRoute and eRoute. The last one is when comparing the performance of the nearest bus lines solution and eRoute, more bus lines that are considered in the subway-bus integrated network would provide more alternative paths and higher supply. However, sometimes, more bus lines would increase the traveling time due to multi-hop transfer. Therefore, in eRoute, we only consider the bus lines that are around the influenced subway line.

Fig. 11 plots the average passenger traveling time of seven solutions over time. The traveling time of a passenger includes the waiting time for the transport service and the actual traveling time. From this Fig. 11, we can see that the average traveling time of eRoute increases from 7:00 to 9:00 because the surge of passenger demand results in the growth of waiting time at the influenced stations. eRoute still outperforms all the other solutions. For example, compared to periodic control with extra vehicles, the passenger traveling time with eRoute is still 20.8 percent less at 9 am. We see the traveling time of all solutions except eRoute decrease at the end of the day, and this is because many passengers arriving at the influenced station after 19:00 do not receive any transport service due to the limited supply, so it only counts their waiting time. The traveling time by

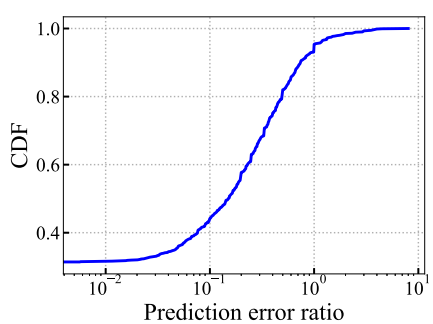


Fig. 9. CDF of prediction error ratio.

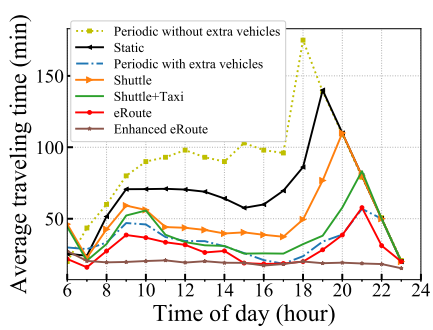


Fig. 11. Average traveling time.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

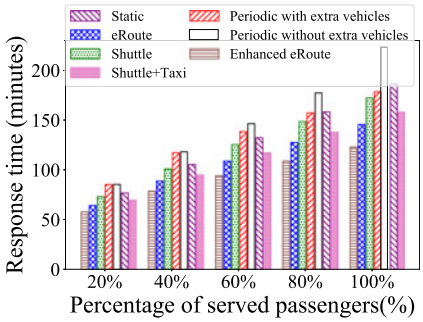


Fig. 12. Response time versus % of served passengers.

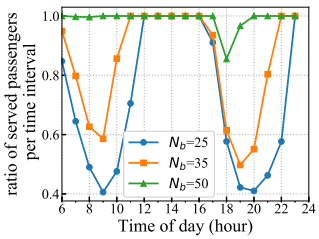


Fig. 13. Impact of extra vehicles.

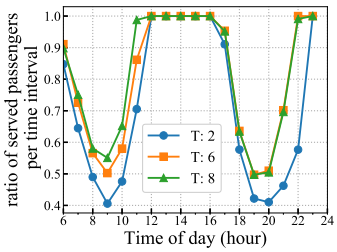


Fig. 14. Impact of horizon length.

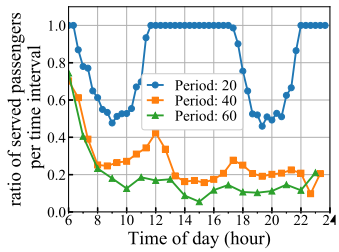


Fig. 15. Impact of update period.

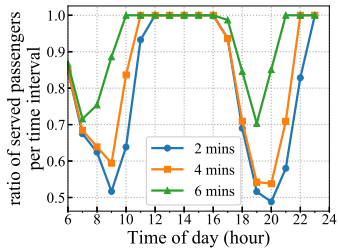


Fig. 16. Impact of detour constraint.

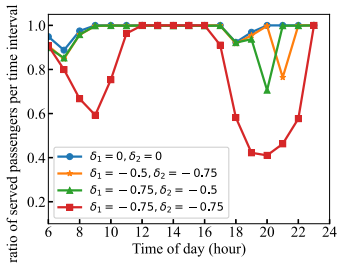


Fig. 17. Performance of enhanced eRoute with different parameters.

enhanced eRoute is stable over the day since most passengers can catch up vehicles without long waiting time when there is enough transport capacity. Here we did not plot the curve of the Nearest solution since its average traveling time is much higher than the others.

Response Time. Once disruptions occur, eRoute and enhanced eRoute reroute nearby buses and taxis to pick up passengers stranded at the influenced subway stations, which has a very short response time. Other solutions like Shuttle or Shuttle+Taxi need to dispatch extra buses from the distant terminals, which usually takes a long time to reach the influenced stations. Fig. 12 shows how long each solution takes to pick up a certain percentage of passengers during the first two-hours after disruptions. We can see that it takes eRoute 88 minutes and Shuttle around 100 minutes to move 40 percent of the passengers, which suggests eRoute has nearly 12.0 percent faster response time to move 40 percent of all the passengers.

Number of Extra Vehicles. We also evaluate the performances of eRoute with a different number of total available buses. As shown in Fig. 13, the more vehicles we use, the higher RSPI we can achieve. When the number of total available extra buses is 25, the RSPI can reach 100 percent during off-peak hours. When 50 buses are used, the RSPI is almost 100 percent in the whole day.

Time Horizon. Fig. 14 plots the performance of eRoute with different prediction time horizon: 2, 6 and 8 time slots. The observation is that during off-peak hours, the percentage of served passengers of 8 time slots horizon outperforms that of 2 and 6 time slots horizon with average gains of 36.2 and 12.3 percent respectively. The reason for this observation is that a shorter time horizon means that only the demand in the very recent future is considered, which misses opportunities to achieve better control. During the other hours of the day, the percentage of served passengers with time horizon 6 and 8 is similar.

Control Update Period. Fig. 15 plots the performance of eRoute with different control update periods: 20, 40 and 60 minutes. The prediction time horizon is set to be 120 minutes. We can see that shorter control update period can increase the performance of eRoute, as it allows more frequent control decisions for passenger demand changes: when the time slot length is 20 minutes, it improves the performance by up to 2.4 and 2.7 times compared with time slot lengths of 40 and 60 minutes respectively at 8:00.

Detour Constraint. Fig. 16 plots the performance of eRoute with different detour constraints: 2, 4, and 6 minutes. The observation is that a higher detour constraint can increase the ratio of served passengers, since it allows more bus lines to be detoured to influenced nodes, and then provides a higher supply for delivering passengers.

Performance of Enhanced eRoute With Different Parameters. Fig. 17 shows the performance of enhanced eRoute with the different settings of δ_1 and δ_2. The two parameters are negative since we want to minimize the idle driving distance

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

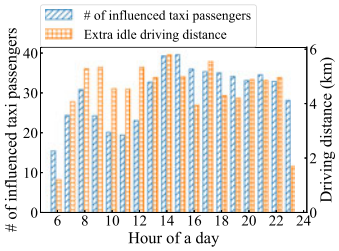


Fig. 18. Cost of enhanced eRoute.

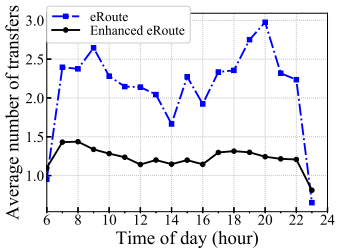


Fig. 19. Overhead of eRoute.

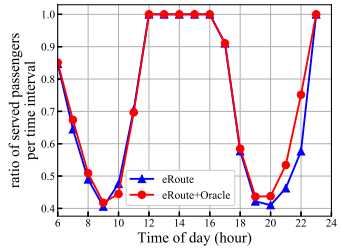


Fig. 20. Performance of eRoute with perfect prediction.

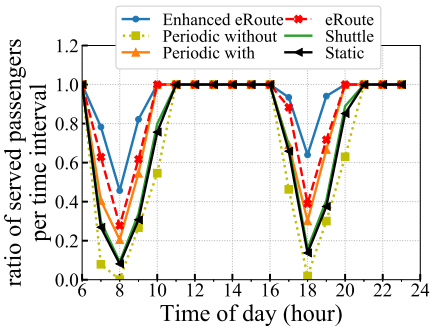


Fig. 21. eRoute under a bus disruption.

and the number of influenced passengers. The main observation is that the performance of enhanced eRoute degrades with the decrease of δ_1 and δ_2. When putting more weight on minimizing the cost of taxis, the number of taxis that are assigned to serve stranded passengers drops in order to affect fewer taxi passengers, or some taxis can only be assigned to the nearby stations or bus lines to reduce the idle driving distance. Therefore, the performance of enhanced eRoute decreases when the two parameters become small.

Cost of Enhanced eRoute. We use the number of influenced taxi passengers and the extra idle driving distance of taxis to measure the cost of enhanced eRoute. The results are shown in Fig. 18. We can observe that only fewer than 40 taxi passengers are influenced by the enhanced eRoute during each time slot, and the number of influenced taxi passengers is high during the morning, afternoon, and evening rush hours. The other observation is that enhanced eRoute does not introduce much extra idle driving distance for taxis, i.e., always less than six kilometers per hour.

Overhead of eRoute. The average number of transfers is used to measure the overhead of eRoute and enhanced eRoute. Since passengers are delivered by rerouted bus lines, which cannot connect origin and destination directly and increases the number of transfers they make. Fig. 19 plots the average number of transfers by eRoute and enhanced eRoute during the day. We can see that passengers need to transfer more than twice to reach their destinations during most time of one day by eRoute. Specifically, during rush hours, more transfers are needed due to higher passenger demand making passengers find the longer alternative path to their destinations. Since taxis provide extra capacity to transport passengers between an influenced station and a bus line passing a station, a fewer number of transfers is required if using enhanced eRoute.

Performance of eRoute With Perfect Prediction of Passenger Demand. Since there exist errors for predicting the future passenger demand as shown in Figs. 8 and 9, we show the performance difference of eRoute with predicted passenger demand and eRoute with actual passenger demand (called eRoute+oracle) in Fig. 20. The main observation is that compared with eRoute+oracle, the performance of eRoute decreases by 3.3 percent on average over the time intervals. We conclude that although the prediction model introduces the error as shown in Figs. 8 and 9, eRoute serves the similar number of passengers with eRoute+oracle does.

Disruption to Bus Systems. To demonstrate the flexibility of our design, we conduct another simulation to evaluate the performance of eRoute under a disruptive failure of the bus system. Here we simulate the transportation network when the bus service is shut down at a few stops in one region of the city due to accidents, and then we apply eRoute to transport stranded bus passengers. The simulated disruption to the bus system influences six stations and 48 bus lines from 6:00 to 23:00. Only the passengers boarding or getting off the bus in one of six bus stops are regarded as being affected. There are around 1630 bus passengers boarding or getting off the bus in one of six bus stops during each 20-minute time slot on average. Existing methods [17], [18] are implemented to estimate the destinations of bus passengers. For the smartcard data that is not in a daily trip chain, the kernel density estimation is used to compute the probability of the destinations.

Fig. 21 shows the RSPI of six solutions that are used to handle a disruptive event to bus systems. We can observe that compared with periodic control with extra vehicles, the ratio of served passengers per time interval of eRoute increases by up to 54.9 percent. The reason for this performance improvement is that eRoute is able to predictively reschedule and reroute nearby bus lines, where periodic control with extra vehicles only reactively adjusts routing and scheduling decisions. It is also observed that more transport capacity is needed to fully serve the stranded passengers during rush hours by comparing eRoute and enhanced eRoute. With more available transportation resources, eRoute will achieve the best performances than the other

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

baselines, because it plans extra vehicles and routes for the dynamic passenger demand after a bus disruption.

7 Discussion

Incorporating Other Factors. (i) The performance of eRoute is impacted by passengers' preference for different paths, which limits the actual number of transported passengers. For instance, some paths with long traveling time or a large number of transfers are not selected by passengers. Therefore, integrating passengers' preference models is a future direction of this work. (ii) Traffic conditions may increase or decrease vehicles' speed, which further affect the transportation capacity. In this study, traffic conditions are related to not only regular vehicles, such as trucks and private cars but also the rescheduled buses and taxis. This work will integrate the traffic condition model in the future. (iii) Modeling the change of passenger demand between any two urban regions due to disruptions is essential to our design. However, since disruptions rarely happen in the city, some data-driven solutions [19], [20] to model passengers' response to disruptions may not work. It is a future study to incorporate the passenger demand change model because of disruptions.

Limitations. The applicability of eRoute is limited by the scale of disruptions and the number of transportation resources. The main idea of eRoute is to use under-utilized transportation resources to transport stranded passengers. However, in preparation for the large-scale urban transportation failure, city authorities need to have sufficient resources, e.g., extra buses, taxis, and trains.

Generalization of eRoute. Although we take the disruptions to transportation systems as an example to design eRoute in this work, the ideas and insights can also be generated to a broader range of urban mobile systems that encounter disruptions. First, complementary urban systems can be coordinated dynamically when the operation of one system is affected by disruptions. Second, according to the characteristics of the given disruptions and urban mobile systems, the dynamic selection of complementary systems is useful for improving integration efficiency. Finally, due to the features of different urban mobile systems, multi-level coordination can reduce the cost, e.g., road congestion around the location where disruptions happen, while increasing the performance of integration.

8 Related Work

Urban Data-Driven Applications and Analysis. There exist some prior works, which either propose data-driven applications or formulate generic models to capture urban phenomena by data analysis. The increasing availability of urban sensors has encouraged a surge of work focusing on design data-driven applications. Many novel applications are proposed to improve the efficiency of the urban transportation system, e.g., balancing bike sharing system [21], providing last-mile transit service to deliver passengers [22], coordinating electric taxis for charging [23], and helping taxi drivers find next passengers efficiently [24]. Based on the collected large-scale data, some works focus on data-driven analysis to formulate generic models to understand urban features, e.g., inferring various traffic indicators [25], inferring human mobility patterns across the city [26], investigating spatiotemporal segmentation information of trips inside a metro system [27], calculating traffic volume on road segments [28], path planning for instant delivery [29] and inferring traffic cascading patterns [30].

Solution to Handle Disruptive Events. They are classified into two directions [7]: pre-disruption preparedness and post-disruption response. *Pre-disruption* preparedness is to prepare certain measures before disruption happens. An alternative direction is designing a robust schedule to enhance potential recovery actions. [31], [32], [33] and [4] consider robust train scheduling. [7] studies that metro network resilience to disruptions can be enhanced by localized integration between bus services and subway stations to achieve the desired resilience to potential disruptions. [34] focuses on how to determine optimal plans for protecting passengers' railway transportation networks under a limited budget. However, their design relies on manual and local incremental adjustments on bus routes, and it generates static and fixed routes. Differently, our solution dynamically adjusts bus routes and based on passenger demand. *Post-disruption* response focuses on coming up with responsive measures for subway system disruptions to alleviate consequences. [35] states that in the case of a disruption, the first task is keeping the subway system running, including timetable adjustment [5], and re-scheduling rolling stock and crew [36]. [37] and [3] introduce shuttle bus services in the disrupted area intelligently which requires extra shuttle buses rather than detouring existing bus lines. [6] has taxis instead of buses as the recovery service for on-board passengers in a public tram system. [38] responds to serious disruptions by redesigning the lines in a particular region around the disruption. [39] proposes an integrated management solution to handle given disruption by rescheduling trains and re-flowing passengers. However, these studies do not consider dynamic data-driven integration among heterogeneous transportation systems.

9 Conclusion

In this work, we study how to address disruptions to urban mobile systems using transportation systems as an example. We design, implement, and evaluate eRoute for dynamic transportation integration under disruptive events based on real-world multi-source data from the city Shenzhen. Our endeavors offer a few valuable insights for fellow researchers to conduct similar investigations: (i) under disruptive events, the existing effort for transfers within public transportation systems provides an opportunity to dynamically integrate them without requiring ad-hoc efforts, e.g., extra bus lines; (ii) given spatial-temporal partitions of public transportation systems and natures of disruptive events, we can deliver stranded passengers with a hierarchical receding horizon control framework to reduce their affected traveling time with minimal overheads; (iii) our work only focuses on the technical frontier on the modeling and resource allocation framework, and it is more challenging to establish working policies that would make large-scale deployment feasible to reduce impacts of disruptive events and increase transportation resilience.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.

ACKNOWLEDGMENTS

This paper was presented in part at the 9th ACM/IEEE International Conference on Cyber-Physical Systems, Porto, Portugal, April 2018 [1]. This research was supported in part by the NSF grants CNS 1553273, CNS 1463722, CPS 1646470 and CNS 1446640, CNS 1952096 and CNS 1553273 (CAREER).

REFERENCES

[1] Y. Yuan, D. Zhang, F. Miao, J. A. Stankovic, T. He, G. Pappas, and S. Lin, "Dynamic integration of heterogeneous transportation modes under disruptive events," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst.*, Apr. 2018, pp. 65–76.

[2] Federal Transit Administration, "Transit asset management practices a national and international review," 2010. [Online]. Available: https://www.transit.dot.gov/sites/fta.dot.gov/files/docs/TAM_A_National_and_International_Review_-_6.10_FINAL_0.pdf

[3] K. Kepaptsoglou and M. G. Karlaftis, "The bus bridging problem in metro operations: Conceptual framework, models and algorithms," *Public Transport*, vol. 1, no. 4, pp. 275–297, 2009.

[4] M. Fischetti, D. Salvagnin, and A. Zanette, "Fast approaches to improve the robustness of a railway timetable," *Transp. Sci.*, vol. 43, no. 3, pp. 321–335, 2009.

[5] C. G. Walker, J. N. Snowdon, and D. M. Ryan, "Simultaneous disruption recovery of a train timetable and crew roster in real time," *Comput. Operations Res.*, vol. 32, no. 8, pp. 2077–2094, 2005.

[6] A. Z. Zeng, C. F. Durach, and Y. Fang, "Collaboration decisions on disruption recovery service in urban public tram systems," *Transp. Res, E, Logistics Transp. Rev.*, vol. 48, no. 3, pp. 578–590, 2012.

[7] J. Jin, L. Tang, L. Sun, and D. Lee, "Enhancing metro network resilience via localized integration with bus services," *Transp. Res., E, Logistics Transp. Rev.*, vol. 63, pp. 17–30, 2014.

[8] N. Garg, V. V. Vazirani, and M. Yannakakis, "Primal-dual approximation algorithms for integral flow and multicut in trees," *Algorithmica*, vol. 18, no. 1, pp. 3–20, 1997.

[9] M. Costa, L. Létocart, and F. Roupin, "Minimal multicut and maximal integer multiflow: A survey," *Eur. J. Oper. Res.*, vol. 162, no. 1, pp. 55–69, 2005.

[10] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems," *J. Comput. Syst. Sci.*, vol. 67, no. 3, pp. 473–496, 2003.

[11] M. A. Fonoberova and D. D. Lozovanu, "The minimum cost multicommodity flow problem in dynamic networks and an algorithm for its solving," *Comput. Sci. J. Moldova*, vol. 13, no. 1, pp. 29–36, 2005.

[12] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *Int. J. Robot. Res.*, vol. 35, no. 1–3, pp. 186–203, 2016.

[13] F. Miao *et al*. "Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 2, pp. 463–478, 2016.

[14] H. Cai, X. Wang, P. Adriaens, and M. Xu, "Environmental benefits of taxi ride sharing in Beijing," *Energy*, vol. 174, pp. 503–508, 2019.

[15] Y. Dong, S. Wang, L. Li, and Z. Zhang, "An empirical study on travel patterns of internet based ride-sharing," *Transp. Res. Part C: Emerg. Technol.*, vol. 86, pp. 1–22, 2018.

[16] J. Jung, R. Jayakrishnan, and J. Park, "Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing," *Comput.-Aided Civil Infrastructure Eng.*, vol. 31, pp. 275–291, 2016.

[17] M. Trépanier, N. Tranchant, and R. Chapleau, "Individual trip destination estimation in a transit smart card automated fare collection system," *J. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 1–14, 2007.

[18] L. He and M. Trépanier, "Estimating the destination of unlinked trips in transit smart card fare data," *Transp. Res. Rec.*, vol. 2535, no. 1, pp. 97–104, 2015.

[19] T. Lin, A. Shalaby, and E. Miller, "Transit user behaviour in response to subway service disruption," 2016. [Online]. Available: https://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=1239&context=csce2016

[20] A. M. Pnevmatikou, M. G. Karlaftis, and K. Kepaptsoglou, "Metro service disruptions: How do people choose to travel?," *Transportation*, vol. 42, no. 6, pp. 933–949, 2015.

[21] S. Wang, T. He, D. Zhang, Y. Liu, and S. H. Son, "Towards efficient sharing: A usage balancing mechanism for bike sharing systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2011–2021.

[22] D. Zhang, J. Zhao, F. Zhang, R. Jiang, and T. He, "Feeder: Supporting last-mile transit with extreme-scale urban infrastructure data," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw.*, 2015, pp. 226–237.

[23] Y. Yuan, D. Zhang, F. Miao, J. Chen, T. He, and S. Lin, "p∧2charging: Proactive partial charging for electric taxi systems," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 688–699.

[24] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 220–232, Jan. 2013.

[25] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Netw.*, vol. 32, no. 4, pp. 40–46, Jul. 2018.

[26] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He, "Exploring human mobility with multi-source data at extremely large metropolitan scales," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 201–212.

[27] F. Zhang, J. Zhao, C. Tian, C. Xu, X. Liu, and L. Rao, "Spatiotemporal segmentation of metro trips using smart card data," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1137–1149, Mar. 2016.

[28] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng, "City-wide traffic volume inference with loop detector data and taxi trajectories," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*. 2017, pp. 1–10.

[29] Y. Zhang *et al*. "Route prediction for instant delivery," *Proc. ACM Interact., Mobile, Wearable, Ubiquitous Technol.*, 2019, vol. 3, pp. 1–25.

[30] Y. Liang, Z. Jiang, and Y. Zheng, "Inferring traffic cascading patterns," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2017, pp. 1–10.

[31] P. J. Zwaneveld, L. G. Kroon, and S. P. Van Hoesel, "Routing trains through a railway station based on a node packing model," *Eur. J. of Oper. Res.*, vol. 128, no. 1, pp. 14–33, 2001.

[32] P. J. Zwaneveld *et al*. "Routing trains through railway stations: Model formulation and algorithms," *Transp. Sci.*, vol. 30, no. 3, pp. 181–194, 1996.

[33] Y. Liu *et al*. "Intelligent bus routing with heterogeneous human mobility patterns," *Knowl. Inf. Syst.*, vol. 50, pp. 383–415, 2017.

[34] S. Starita and M. Paola Scaparra, "Passenger railway network protection: A model with variable post-disruption demand service," *J. Oper. Res. Soc.*, vol. 69, no. 4, pp. 603–618, 2018.

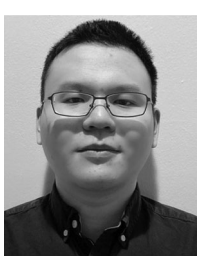
[35] L. Kroon and D. Huisman, "Algorithmic support for railway disruption management," in *Proc. Transitions Towards Sustain. Mobility*, 2011, pp. 193–210.

[36] L. Cadarso, Á. Marín, and G. Maróti, "Recovery of disruptions in rapid transit networks," *Transp. Res., E, Logistics Transp. Rev.*, vol. 53, pp. 15–33, 2013.

[37] C. Luo *et al*. "Data-driven disruption response planning for a mass rapid transit system," in *Proc. Smart Transp. Syst.*, 2019 pp. 205–213.

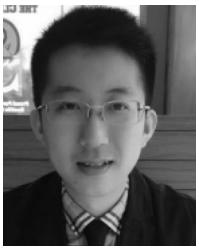
[38] A. Kiefer, S. Kritzinger, and K. F. Doerner, "Disruption management for the Viennese public transport provider," *Public Transport*, vol. 8, no. 2, pp. 161–183, 2016.

[39] N. Bešinović, Y. Wang, S. Zhu, E. Quaglietta, T. Tang, and R. M. P. Goverde, "Integrated train and passenger disruption management for urban railway lines," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, 2019, pp. 3182–3187.



Yukun Yuan (Student Member, IEEE) received the BE degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2015. He is currently working toward the PhD degree at the Department of Electrical and Computer Engineering, Stony Brook University. His research interests include data-driven urban cyber-physical systems, including transportation system integration under disruptions and electric taxis charging scheduling.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.



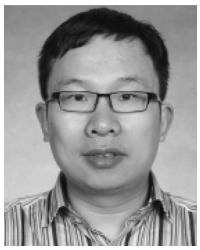
Desheng Zhang (Member, IEEE) is currently an assistant professor at the Department of Computer Science, Rutgers University, New Brunswick, NJ, USA. His research interests include bridging cyber-physical systems and big urban data by technical integration of communication, computation, and control in data-intensive urban systems, the life cycle of Big Data-driven urban systems, from multisource data collection to streaming-data processing, heterogeneous data management, model abstraction, visualization, privacy, service design, and deployment in complex urban settings, and real-time interactions among heterogeneous urban systems including cellphone, smartcard, taxi, bus, truck, subway, bike, personal vehicle, electric vehicle, and road networks.



Fei Miao (Member, IEEE) received the BSc degree in automation from Shanghai Jiao Tong University, in 2010, and the MA degree in statistics and the PhD degree in electrical and systems engineering from the University of Pennsylvania, in 2015 and 2016, respectively. She is currently an assistant professor at the Department of Computer Science and Engineering, University of Connecticut. She was a postdoc researcher with the Department of Electrical and Systems Engineering, GRASP Laboratory and PRECISE center, University of Pennsylvania. Her research interests include data-driven real-time optimization and control of cyber-physical systems under model uncertainties, and resilient and secure control cyber-physical systems. She was the recipient of the Charles Hallac and Sarah Keil Wolf Award for Best Doctoral Dissertation from the University of Pennsylvania and a Best Paper Award finalist at the 6th ACM/IEEE International Conference on Cyber-Physical Systems in 2015.



John A. Stankovic (Life Fellow, IEEE) received the PhD degree from Brown University, Providence, RI, USA. He is currently the BP America professor at the Computer Science Department, University of Virginia, Charlottesville, VA, USA. He has an H-index of 116 and more than 61,000 citations. He is a fellow of the ACM. He was the recipient of the Honorary Doctorate from the University of York, the IEEE Real Time Systems Technical Committee's Award for outstanding technical contributions and leadership, the IEEE Technical Committee on Distributed Processing's Distinguished Achievement Award, and seven best paper awards, including one for ACM SenSys 2006.



Tian He (Fellow, IEEE) is currently a professor at the School of Computer Science and Engineering, Southeast University. He has authored or coauthored more than 300 articles in premier network journals and conferences with more than 23,000 citations and an H-Index 70. His research interests include wireless networks, networked sensing systems, cyber-physical systems, real-time embedded systems, and distributed systems. He is a fellow of ACM. He was the recipient of the NSF CAREER Award in 2009, the McKnight Land-Grant Chaired Professorship in 2011, the George W. Taylor Distinguished Research Award in 2015, the China NSF Outstanding Overseas Young Researcher I and II in 2012 and 2016, and eight Best Paper Awards in international conferences, including MobiCom, SenSys, and ICDCS.



George J. Pappas (Fellow, IEEE) received the PhD degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998. He is currently the UPS foundation professor and chair of the Department of Electrical and Systems Engineering, University of Pennsylvania. He was the deputy dean for research with the School of Engineering and Applied Science. His research interests include control theory and in particular, hybrid systems, embedded systems, hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks. He was the receipient of various awards such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, the O. Hugo Schuck Best Paper Award, the National Science Foundation PECASE, and the George H. Heilmeier Faculty Excellence Award.



Shan Lin (Senior Member, IEEE) received the PhD degree in computer science from the University of Virginia, Charlottesville, VA, USA, in 2010. His PhD dissertation was on taming networking challenges with feedback control. He is currently an associate professor at the Electrical and Computer Engineering Department, Stony Brook University, Stony Brook, NY, USA, where he is involved in wireless network protocols, interoperable medical devices, smart transportation systems, and intelligent sensing systems. His current research interests include networked systems, with an emphasis on feedback control-based design for cyber-physical systems and sensor systems. He was a recipient of the NSF CAREER Award in 2016.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:35:45 UTC from IEEE Xplore. Restrictions apply.